

WHAT IS CLAIMED IS:

1 1. A keystore method comprising the steps of:
2 retrieving one or more certificates from a local database;
3 determining if said any of said one or more certificates preexists in a preselected
4 portion of a distributed database; and
5 storing nonpreexisting certificates of said one or more certificates in said
6 preselected portion of said distributed database.

1 2. The method of claim 1 wherein said preselected portion of said distributed
2 database comprises said distributed database.

1 3. The method of claim 1 further comprising the step of determining if said one or
2 more certificates is invalid.

1 4. The method of claim 3 wherein said step of storing nonpreexisting ones of said
2 one or more certificates is bypassed for invalid certificates.

1 5. The method of claim 3 further comprising the step of requesting a new certificate
2 corresponding to an invalid certificate.

1 6. The method of claim 1 further comprising the step of updating said distributed
2 database in response to an update event.

1 7. The method of claim 6 wherein said step of updating said distributed database
2 comprises the steps of:

3 requesting one or more new certificates; and
4 adding said new certificates to said distributed database.

1 8. The method of claim 1 further comprising the steps of:
2 determining if a current certificate supercedes a preexisting certificate; and
3 replacing said preexisting certificate with said current certificate if said current
4 certificate supercedes said preexisting certificate.

1 9. The method of claim 1 further comprising the steps of:
2 accessing said distributed keystore; and
3 requesting a selected certificate from said distributed keystore.

1 10. The method of claim 9 further comprising the step of searching a local keystore
2 for said selected certificate in response to a failure of said step of requesting said selected
3 certificate.

1 11. The method of claim 1 further comprising the step of repeating, for a second local
2 database, the steps of:

3 retrieving one or more certificates;
4 determining if said any of said one or more certificates preexists in a preselected
5 portion of a distributed database; and
6 storing nonpreexisting certificates of said one or more certificates in said
7 preselected portion of said distributed database.

1 12. The method of claim 8 wherein said distributed database comprises a logical
2 keystore.

1 13. A computer program product embodied in a tangible storage medium, the
2 program product for managing a keystore, the program product including a program of
3 instructions for performing the steps of:

4 retrieving one or more certificates from a first local database;
5 determining if said any of said one or more certificates preexists in a preselected
6 portion of a distributed database; and
7 storing nonpreexisting certificates of said one or more certificates in said
8 preselected portion of said distributed database.

1 14. The program product of claim 13 wherein said preselected portion of said
2 distributed database comprises said distributed database.

1 15. The program product of claim 13 wherein said program of instructions further
2 comprises programming for performing the step of determining if said one or more
3 certificates is invalid.

1 16. The program product of claim 15 wherein said step of storing nonpreexisting
2 ones of said one or more certificates is bypassed for invalid certificates.

1 17. The program product of claim 15 wherein said program of instructions further
2 comprises programming for performing the step of requesting a new certificate
3 corresponding to an invalid certificate.

1 18. The program product of claim 13 wherein said program of instructions further
2 comprises programming for performing the step of updating said distributed database in
3 response to an update event.

1 19. The program product of claim 18 wherein said step of updating said distributed
2 database comprises the steps of:
3 requesting one or more new certificates; and
4 adding said new certificates to said distributed database.

1 20. The program product of claim 13 wherein said program of instructions further
2 comprises programming for performing the steps of:
3 determining if a current certificate supercedes a preexisting certificate; and
4 replacing said preexisting certificate with said current certificate if said current
5 certificate supercedes said preexisting certificate.

1 21. The program product of claim 13 wherein said program of instructions further
2 comprises programming for performing the steps of:
3 accessing said distributed database; and
4 requesting a selected certificate from said distributed database.

1 22. The program product of claim 21 wherein said program of instructions further
2 comprises programming for performing the step of searching a local keystore for said
3 selected certificate in response to a failure of said step of requesting said selected
4 certificate.

1 23. The computer program product of claim 13 wherein said program of instructions
2 further comprises instructions for the step of repeating, for a second local database, the
3 steps of:

4 retrieving one or more certificates;

5 determining if said any of said one or more certificates preexists in a preselected
6 portion of a distributed database; and

7 storing nonpreexisting certificates of said one or more certificates in said
8 preselected portion of said distributed database.

1 24. The computer program product of claim 20 wherein said distributed database
2 comprises a logical keystore.

1 25. A data processing system comprising:
2 circuitry operable for retrieving one or more certificates from a first local
3 database;
4 circuitry operable for determining if said any of said one or more certificates
5 preexists in a preselected portion of a distributed database; and
6 circuitry operable for storing nonpreexisting certificates of said one or more
7 certificates in said preselected portion of said distributed database.

1 26. The system of claim 25 wherein said preselected portion of said distributed
2 database comprises said distributed database.

1 27. The system of claim 25 further comprising circuitry operable for determining if
2 said one or more certificates is invalid.

1 28. The system of claim 27 wherein said circuitry operable for determining if said one
2 or more certificates is expired includes circuitry operable for bypassing, for invalid
3 certificates, said circuitry operable for storing nonpreexisting certificates.

1 29. The system of claim 27 further comprising circuitry operable for requesting a new
2 certificate corresponding to an invalid certificate.

1 30. The system of claim 25 further comprising circuitry operable for updating said
2 distributed database in response to an update event.

1 31. The system of claim 30 wherein said circuitry operable for updating said
2 distributed database comprises:

3 circuitry operable for requesting one or more new certificates; and
4 circuitry operable for adding said new certificates to said distributed database.

1 32. The system of claim 25 further comprising:
2 circuitry operable for determining if a current certificate supercedes a preexisting
3 certificate; and
4 circuitry operable for replacing said preexisting certificate with said current
5 certificate if said current certificate supercedes said preexisting certificate.

1 33. The system of claim 25 further comprising:
2 circuitry operable for accessing said distributed database; and
3 circuitry operable for requesting a selected certificate from said distributed
4 database.

1 34. The system of claim 33 further comprising circuitry operable for searching a local
2 keystore for said selected certificate in response to a failure of said step of requesting said
3 selected certificate.

The $\{b_{\alpha\beta}^{(i)}\}$ are given by (2.1) with α, β running over the $2n$ components of \mathbf{b} . The $\{b_{\alpha\beta}^{(i)}\}$ are given by (2.1) with α, β running over the $2n$ components of \mathbf{b} . The $\{b_{\alpha\beta}^{(i)}\}$ are given by (2.1) with α, β running over the $2n$ components of \mathbf{b} .

- 1
- 2
- 3
- 4
- 5
- 6
- 7

4
5

6
7

1
2